

“十三五”高校计算机应用技术系列规划教材

丛书主编 谭浩强

C语言程序设计

(第四版)

C Yuyan Chengxu Sheji

林小茶 编著

谭浩强 主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十三五”高校计算机应用技术系列规划教材

丛书主编 谭浩强

C 语言程序设计

(第四版)

林小茶 编著

谭浩强 主审

内 容 简 介

C 语言是程序员的入门语言，也是许多大学为学生安排的第一门程序设计课程。本书充分考虑到这一点，在内容的编排上尽量符合初学者的要求，在实例的选择上从易到难，循序渐进，并且能够解决一些实际问题。

本书主要内容包括：C 语言概述、C 语言的基本知识、顺序和选择结构程序设计、循环结构程序设计、函数、数组、指针、结构体等构造数据类型、文件和综合案例。全书通过大量的实例讲解了用 C 语言进行结构化程序设计的要领。

为了方便读者参加全国计算机等级考试二级 C 语言程序设计的考试，本版教材将最新的考试题目分门别类，按照对应的学习内容在第 2 章~第 8 章的最后一节做了讲解。另外，本书还配有上机指导书，给出了本书部分编写程序题目的参考答案，供广大教师和学生参考。

本书适合作为大学应用型本科院校的学生学习 C 程序设计课程的教材，也可作为 C 语言自学者的教材或参考书。

图书在版编目 (CIP) 数据

C 语言程序设计 / 林小茶编著. —4 版. —北京:
中国铁道出版社, 2016. 6

“十三五”高校计算机应用技术系列规划教材

ISBN 978-7-113-21462-3

I. ①C… II. ①林… III. ①C 语言—程序设计—高等
学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 027732 号

书 名: C 语言程序设计 (第四版)

作 者: 林小茶 编著

策 划: 周海燕

读者热线: 010 (63550836)

责任编辑: 周海燕 彭立辉

封面设计: 付 巍

封面制作: 白 雪

责任校对: 汤淑梅

责任印制: 郭向伟

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 中国铁道出版社印刷厂

版 次: 2004 年 5 月第 1 版 2007 年 7 月第 2 版 2010 年 12 月第 3 版 2016 年 6 月第 4 版
2016 年 6 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 20.5 字数: 486 千

书 号: ISBN 978-7-113-21462-3

定 价: 43.00 元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：(010) 63550836

打击盗版举报电话：(010) 51873659

“十三五”高校计算机应用技术系列规划教材

主任：谭浩强

副主任：陈维兴 严晓舟

委员：(按姓氏音序排列)

安淑芝	安志远	陈志泊	韩 劼	侯冬梅
李 宁	李雁翎	林成春	刘宇君	秦建中
秦绪好	曲建民	尚晓航	邵丽萍	宋 红
宋金珂	王兴玲	魏善沛	熊伟建	薛淑斌
张 玲	赵乃真	訾秀玲		

信息技术的迅猛发展和对人类的深远影响使许多人目瞪口呆。在当今社会，每个人都在享受信息技术的成果，都在直接或间接地应用着信息技术。信息技术改变了世界面貌，改变了人类的生活方式，也改变了人们的思维方式。

早在 30 多年前，我国高等学校已经开始在全体大学生中开展计算机教育，计算机课程成为所有学生的必修课程，掌握计算机基本知识和应用能力成为对大学生的基本要求和毕业后求职的必要条件。大学中的计算机课程的设置和内容随着信息技术的发展而与时俱进，全社会对计算机基础教育的认识和支持率大大提高了，真是今非昔比。

高等学校中的计算机教育是由两部分组成的：一是计算机专业的教育，二是面向 95% 以上大学生的非计算机专业的计算机教育（称为高校计算机基础教育）。两者的培养目标、教学内容和教学方法是不同的。前者主要培养计算机专门人才，后者主要培养各行各业中广大的计算机应用人才。

过去，面向非计算机专业大学生的课程体系和教材往往是根据计算机专业的知识体系和教材来构建的，强调学科的系统性和完整性，强调理论，有的甚至是计算机专业课程的浓缩。事实证明这是不切合实际的，难以取得好的效果。

大学生为什么要学习计算机？答案是不言而喻的：首先是因为计算机有用。如果没有用何必学习它呢？现代社会离开计算机寸步难行，使用计算机将是现代人的一项基本技能。现在有些老年人（包括一些老年知识分子）由于不会使用计算机而感到处处不便，他们的意识、习惯和工作明显落后于时代，影响了他们对社会的贡献，这是很可惜的。

有人轻视应用，以为应用就是操作，因此认为“理论高级，应用低级”，这是一种误解。应用是分层次的，应用有初级、中级和高级之分。搞理论的人只是少数，绝大多数人将来是搞应用的。大到两弹一星，小到网上购物，在各个领域，都可以看到计算机应用无所不在，所有的人都可以尽其所能，大显身手。

计算机基础教育在本质上是计算机应用的教育，应当以应用为目的，以应用为出发点，应该以计算机应用为主线来构建课程体系，明确分析和提出应用能力的要求，编写出体现应用特点的教材。

计算机基础教育要综合考虑三方面因素：信息技术的发展、面向应用的需要以及科学思维的培养。在计算机基础教学中应当做到：讲知识、讲应用、讲方法，并且把三者紧密结合起来。首先要讲知识，因为知识是基础，应用和方法都需要知识支撑；同时要讲应用，因为计算机基础教育不是纯理论的学习，要面向应用，提高应用能力；还要注意向学生传授方法，使学生掌握规律，学会思考，培养科学的思维方法。

对大多数人来说，学习计算机的目的是利用这个现代化工具处理面临的各种问题，使自己能够跟上时代前进的步伐，同时在学习过程中努力培养自己的信息素养，使自己具有信息时代所要求的科学素质，站在信息技术发展和应用的前列，推动我国信息技术的发展。

学习计算机课程有两种不同的方法：一是从理论入手；二是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说，不仅需要“知道什么”，更重要的是“会做什么”。因此，在学习过程中要以应用为目的，注重培养应用能力，大力加强实践环节训练，激励创新意识。

由于全国各地、各高等院校的情况不同，需要有不同特点的教材来满足不同学校、不同专业教学的需要。因此，在教材建设上应当提倡百花齐放，推陈出新。应当提供不同内容、不同风格的教材，供各校选用。

根据培养应用型人才的需要，我们组织编写了这套“‘十三五’高校计算机应用技术系列规划教材”。这套教材的特点是突出应用技术，面向实际应用，强调培养应用能力，学以致用。在选材上，根据实际应用的需要决定内容的取舍，重视实践环节，不涉及过多的理论，坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上，采取“提出问题—解决问题—归纳分析”的三部曲，这种从实际到理论、从具体到抽象、从个别到一般的方法，符合人们的认知规律，且在实践过程中已取得了很好的效果。

本丛书可以作为应用型大学的计算机应用技术课程的教材，程度较高的高职高专学校也可从中选择适用的教材，也可作为广大计算机爱好者的自学教材。

本丛书由浩强创作室与中国铁道出版社共同策划，由有丰富教学经验的高校老师编写而成。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版推广过程中，得到各高等院校老师的大力支持，对此谨表衷心的感谢。

本丛书如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本丛书的出版，能为我国计算机教育事业的发展和人才培养做出贡献。

全国高等院校计算机基础教育研究会荣誉会长
丛书主编

谭浩强

本书的前三版已经印刷了多次，此次进行第四次修订。

首先感谢谭浩强老师对本书进行了审校，并提出了宝贵意见。根据谭浩强老师的建议，本书在修订的过程中将“函数说明”修改为“函数声明”，全局变量等存储类别的“变量说明”修改为“变量声明”，还包括对结构体类型的“声明”等内容都做了适当的修改。另外，关于“字符串变量”的提法谭老师也提出了中肯的意见，本书采用了字符数组提法。第三，关于调试程序的版本，谭老师建议使用 Visual Studio 2010，因此本书和配套辅导教材以 Visual Studio 2010 作为调试工具，并对调试的过程和方法进行了说明。第四，尽量符合 C99 的标准，例如注释符全部改为“//”。

本书延续了前三版的特点，并在此基础上对所讲解的内容做了适当修订，以适合第一次学习程序设计者的需求。

主要的修订内容包括：

(1) 选用了一些新的案例。这些案例不但更适合初学者，也更有实际意义。例如，有关人的姓名在计算机内的存储。

(2) 教学内容的先后做了部分调整。尤其是第 2 章，因为该章的内容比较烦琐，学生也觉得枯燥，重新调整后更利于教学，教师在使用时也可以根据情况增删该部分内容。第 6 章将二维数组放到了最后，将字符数组和字符串的内容提前了。

(3) 增加了第 10 章综合案例。尽管前 9 章的一些案例随着教学进程不断丰富，已经比较复杂，但是笔者还是认为不够，为了让学生学到更多的程序设计案例，增加了第 10 章，希望对大家有更大的帮助。

(4) 将大部分程序的架构改为

```
int main()
{
    ...
    return 0;
}
```

习题中加“*”号的表示老师可以留作业，习题解答中未给出答案。

本书能够多次再版，要感谢所有使用本书作为教材和学习参考书的教师和读者，同时再次殷切希望读者对本书的内容和编写方法提出宝贵的意见和建议。

由于编者水平有限，疏漏与不妥之处在所难免，敬请广大读者批评指正。

编者

2015年9月

第三版前言

FOREWORD

本教材的第一版和第二版已经印刷了多次，根据读者的需求，本书进行了第三次修订。

本书延续了前两版的特点，对所讲解的内容做了一些选择，以适合第一次学习程序设计者的需求。

第一，在内容的选择上更适合初学者的需求。

例如，链表是 C 程序设计的难点，许多学生都反映链表的程序不能理解，因此，这里讲清楚单链表的基本概念和操作要点，让学生对链表有个基础的了解，而比较深入的内容留到《数据结构》这门课程；又如，讲解了位运算的 C 程序设计的另外一个难点，但是如果学过《汇编语言》，这部分内容是很容易掌握的；再如，用文本方式显示数据的目的只是为了给调试程序带来方便，花大量的精力学习显示格式是没有必要的。

第二，强调好的程序设计思想，不能拘泥于小的程序设计技巧，而破坏了整个程序的清晰度。

例如，本书将告诉学生避免使用像“i+++++i”这样的表达式，而不是花大量的篇幅去分析这个表达式到底等于多少，因为这是毫无意义的，资深程序员建议，使用++运算符的最好方法是只使用单句“i++;”，绝不能将++符号使用在一个复杂的表达式里面。类似地，本书在介绍运算符的优先级和结合性时也提出使用括号表示运算的先后是最好的方法。

第三，编写一些比较实用的程序。

本书中有一个例子随着学习内容的深入，不断地被扩充功能，在选择结构程序设计中只有几行语句，而到文件处理时，这个例子已经被扩充到了 100 行以上。这些程序几乎用到了所有我们学过的基本数据类型和构造数据类型，笔者希望读者通过对这些程序的跟踪分析，提高程序设计的能力。

第四，抛弃那些与建立好的程序设计思想关系不大的内容，选择了更具特色和实用性的程序作为实例。

例如，奥运奖牌榜、按新的个人所得税纳税方式计算纳税额等等。

第五，沿用了第二版中好的写作思想。

在提出问题的同时给出示例程序，而示例程序中尽量将主要的知识点演示出来，使读者对解决同类问题的程序设计思想有比较全面的认识。在示例程序之后，才是对具体问题的讲解和讨论。

第六，对前两版的写作内容进行了重新整合。

在前两版的教材中，按照惯例将 C 语言的基础知识分为两章讲解，但是在实际应用中发现有些问题，就是开始学习的简单知识偏多，学生感觉有些无聊，因为讲解这些知识的案例几乎没有实用价值，主要是在讨论 C 语言的语法问题，而不是解决问题的程序设计思想。因此，在本版教材中将必需的基础知识安排在第 2 章讲解，而将其他的基础知识与后续内容相结合，力争实例更合理。这是一次尝试，希望得到大家的认可，也希望大家提出宝贵意见。

第七，在第 2 章到第 8 章增加了一节内容“典型错误及典型例题”。

典型错误汇集了作者在教学过程中发现的学生最容易犯的错误，并讲解了正确的方法，典型

例题则是取自最近的一次全国计算机等级考试二级 C 语言程序设计的试卷，通过分析考点给出正确答案，希望为读者参加该考试奠定良好的基础。

本书能够多次再版，要感谢所有使用本书作为教材和学习参考书的教师和读者，同时再次殷切希望您对本书的内容和编写方法提出宝贵的意见和建议。感谢杨铮、王昊、赵骏、贾轩、刘宗汛、陈宇航、李月阳、刘博、杨牧天、孙顺旌等同学帮忙调试了全部程序。

由于编者水平有限，疏漏在所难免，请广大读者批评指正。

编 者

2010 年 4 月

第二版前言

FOREWORD

本教材由于定位比较准确，受到使用者的欢迎，已经重印了5次。作者总结了近几年的教学经验，并听取了专家和读者的意见，对本书做了进一步的修订。

这次修订不但在内容上做了一些必要的更新，在写作思想上也做了一些改进，希望能满足初学者的需要。

这种新的写作思想是：在提出问题的同时给出示例程序，而示例程序中尽量将主要的知识点演示出来，使读者对解决同类问题的程序设计思想有比较全面的认识。在示例程序之后，才是对具体问题的讲解和讨论。

同时，本书在以下几方面对第一版的内容做了修订。

(1) 增加了一些能提高程序设计能力的内容，而删减了与建立程序设计思想关系不大的内容。例如，增加了“控制循环的实用方法”一节，而减少了“赋值表达式的类型转换”的内容。

(2) 增加了对程序实例的分析和讲解。本书的一个特点是实用程序较多，但是，由于课时的限制，教师在课上不能将所有程序都讲解得非常全面，而初学者自己阅读又有一定的困难，因此增加了这部分内容。

(3) 更换了一些程序实例，选择了更具特色和实用性的程序实例。例如，奥运奖牌榜、按新的个人所得税纳税方式计算纳税额等等。

(4) 增加了一部分习题，并且这部分习题将不会在配套的习题解答中出现，希望学生能发挥自己的主观能动性学好程序设计。

(5) 本书的全部程序都重新在 Visual C++ 6.0 环境下进行了调试，与调试环境有关的内容也改为 Visual C++ 6.0 的，以适应新的教学需求。并且，除了中文显示有些问题，所有程序依然能在 Turbo C++3.0 环境下运行，以方便那些习惯使用 Turbo C++3.0 环境的师生。

最后，借此次本书再版的机会，向使用本书作为教材和学习参考书的教师和读者表示衷心的感谢，并殷切希望您对本书的内容和编写方法提出宝贵的意见和建议。

由于编者水平有限，疏漏在所难免，请广大读者批评指正。

编者

2007年4月

对于从未接触过程序设计语言的人来说,以 C 语言作为入门也许是有了一定难度,因为 C 语言的语法比较灵活,有些在其他高级语言(例如 Pascal)中被认定具有编译错误的语句,可以轻而易举地通过 C 编译,但是程序运行的结果却不能达到程序设计者的要求,使学习者觉得 C 语言很难。笔者有多年 C 语言教学的经验,对此感受颇深,大约十年以前,在大学教学中, C 语言一般作为第二门甚至第三门程序设计语言课程,所以,学生学习起来是很轻松的,如果教师不能讲一些有深度的内容,反而会使学生觉得教师水平太低。所以, C 语言作为入门语言,其侧重点应有所不同。

首先,在内容的选择上不用做到面面俱到。例如,链表的操作不用讲太多,只要讲清楚单链表的基本概念和操作要点就行了,更多的内容应该交给《数据结构》这门课程;又例如,有关位操作的程序设计,你就是给学生讲了,他们也很难搞清楚,如果学过《汇编语言》,他们会很容易学会这部分内容;再比如,用文本方式显示数据的目的是为了调试程序,花大量的精力学习显示格式是没有必要的。因此,本书在编排上,做了一些小的改革。

第二,要强调好的程序设计思想,不要拘泥于小的技巧,而破坏了整个程序的清晰度。例如,本书将告诉学生避免使用像“`i+++++i`”这样的表达式,而不是花大量的篇幅去分析这个表达式到底等于多少,因为这是毫无意义的,资深程序员建议,使用++符号的最好方法是只使用单句“`i++;`”,绝不能将++符号使用在一个复杂的表达式里面。类似地,本书在介绍运算符的优先级和结合性时也提出使用括号表示运算的先后是最好的方法。

第三,编写一些比较实用的程序。本书中有一个例子随着学习内容的深入,不断地被扩充功能,在选择结构程序设计中只有几行语句,而到文件处理时,这个例子已经被扩充到了 100 行以上。这些程序几乎用到了所有我们学过的基本数据类型和构造数据类型,笔者希望读者通过对这些程序的跟踪分析,提高程序设计的能力。

本书在编写过程中,尽量使内容通俗易懂,适于自学,由浅入深,便于理解。

作为本书的姐妹篇,我们将同时出版本教程的习题解答和实验指导书,给出本教材中所有习题的参考答案,供读者学习时借鉴和参考。

在本书编写和出版过程中,全国计算机基础教育研究会理事长谭浩强教授给予了指导和把关,在此表示衷心的感谢。

在本书的编写和出版过程中还得到了陈维兴教授的帮助和支持,在此表示诚挚的感谢。

编者水平有限,错误在所难免,请广大读者批评指正。

编者

2004 年 4 月

目录

CONTENTS

第 1 章 C 语言概述	1	2.6 不同数据类型的输入/输出	23
1.1 程序与程序设计语言	1	2.6.1 整型数据的输入/输出	23
1.1.1 程序	1	2.6.2 浮点数据的输入/输出	24
1.1.2 程序设计语言	2	2.6.3 字符数据的输入/输出	25
1.2 C 语言发展历史和主要特点	3	2.6.4 用 getchar()输入字符和用 putchar()输出字符	26
1.2.1 C 语言的发展历史	3	2.7 不同数据类型数据间的混合运算	27
1.2.2 C 语言的主要特点	4	2.7.1 自动转换	28
1.3 C 语言的基本结构	4	2.7.2 强制转换	28
1.3.1 第一个程序	5	2.7.3 安全类型转换与非安全类型 转换	29
1.3.2 第二个程序	5	2.8 字符集和标识符	30
1.3.3 printf()使用初步	6	2.8.1 字符集	31
1.3.4 第三个程序	7	2.8.2 标识符	31
1.4 程序的调试	7	2.9 典型错误及典型例题	32
1.4.1 调试步骤	7	小结	35
1.4.2 在 Visual Studio 2010 调试 环境下调试第一个程序	8	习题	36
小结	12	第 3 章 顺序和选择结构程序设计	37
习题	12	3.1 结构化程序设计	37
第 2 章 C 语言的基本知识	13	3.1.1 结构化程序设计的 3 种 基本结构	37
2.1 变量	13	3.1.2 结构化程序设计思想的 产生	42
2.2 常量	15	3.2 语句与分程序	44
2.3 C 语言的数据类型	16	3.3 顺序结构程序设计	45
2.3.1 为什么要讨论数据类型	16	3.4 算术运算符与赋值运算符	46
2.3.2 C 语言有哪些数据类型	17	3.4.1 算术运算符的种类及运算	46
2.3.3 基本数据类型	17	3.4.2 算术表达式及算术运算符 的优先级	47
2.4 不同数据类型的变量说明	18	3.4.3 算术运算符的结合性	48
2.4.1 整型变量	18	3.4.4 普通赋值运算符	48
2.4.2 浮点变量	19	3.4.5 复合赋值运算符	49
2.4.3 字符变量	20	3.5 选择结构程序设计	49
2.5 不同数据类型的常量写法	20		
2.5.1 整型常量	20		
2.5.2 浮点常量	21		
2.5.3 字符常量	21		

3.5.1	问题提出与程序示例	49	4.7.1	continue 的用法	96
3.5.2	if 形式	51	4.7.2	break 与 continue 的区别	96
3.5.3	if...else 形式	53	4.8	程序举例	97
3.5.4	else if 形式	55	4.9	典型错误及典型例题	104
3.5.5	嵌套的 if 语句	58	小结	108	
3.6	关系运算符与逻辑运算符	60	习题	108	
3.6.1	关系运算符	60	第 5 章 函数	111	
3.6.2	逻辑运算符	61	5.1	问题提出与程序示例	111
3.7	增 1/减 1 运算符	62	5.2	函数基础	112
3.8	求字节数运算符	63	5.3	函数的定义	113
3.9	switch 语句	65	5.3.1	函数的定义形式	113
3.10	条件运算符	68	5.3.2	函数的返回值	114
3.11	程序举例	70	5.4	函数调用	115
3.12	典型错误及典型例题	73	5.4.1	函数的调用方式	115
小结	78		5.4.2	嵌套调用	116
习题	79		5.5	函数声明	118
第 4 章 循环结构程序设计	81		5.6	参数传递	119
4.1	问题提出与程序示例	81	5.6.1	形参和实参	119
4.2	while 语句	83	5.6.2	形参的数据类型是基本数据类型	120
4.2.1	while 语句的语法和流程图	83	5.7	递归调用	121
4.2.2	使用 while 语句需要注意的问题	84	5.8	变量的存储类别	126
4.3	do...while 语句	86	5.8.1	自动变量与外部变量	126
4.3.1	do...while 语句的语法和流程图	86	5.8.2	静态变量	130
4.3.2	使用 do...while 语句解决问题	86	5.8.3	寄存器变量	132
4.3.3	使用 do...while 语句需要注意的问题	88	5.9	典型错误及典型例题	133
4.4	for 语句	89	小结	136	
4.4.1	for 语句的语法和流程图	89	习题	137	
4.4.2	使用 for 语句解决问题	90	第 6 章 数组	138	
4.4.3	使用 for 语句需要注意的问题	91	6.1	问题提出与程序示例	138
4.5	多重循环	92	6.2	一维数组	140
4.6	break 语句在循环语句中的用法	94	6.2.1	一维数组的定义	140
4.7	continue 语句	96	6.2.2	一维数组的引用	141
			6.2.3	一维数组的初始化	141
			6.2.4	程序举例	142
			6.3	数组作为函数的参数	146
			6.4	字符串与字符串函数	149

6.4.1	字符数组	149	7.6.1	问题提出与程序示例.....	203
6.4.2	使用字符串常量初始化 字符数组	150	7.6.2	指针数组的定义和使用...	204
6.4.3	对字符数组进行输入/ 输出操作	150	7.7	指针与二维数组	207
6.4.4	字符串函数.....	153	7.7.1	用指针方法操作二维数组	207
6.4.5	程序举例	157	7.7.2	动态的二维数组	209
6.5	二维数组	160	7.7.3	用指向数组的指针操作 二维数组	211
6.5.1	二维数组 OP 的定义.....	160	7.8	命令行参数.....	213
6.5.2	二维数组的引用	161	7.9	典型错误及典型例题	217
6.5.3	二维数组的初始化	162	小结	223	
6.5.4	程序举例	162	习题.....	224	
6.6	典型错误及典型例题	168	第 8 章 结构体等构造数据类型	226	
小结	174	8.1	结构体	226	
习题	174	8.1.1	问题提出与程序示例.....	226	
第 7 章 指针	178	8.1.2	结构体的声明和结构体 变量的定义.....	227	
7.1	指针类型与指针运算符.....	178	8.1.3	结构体成员的引用	228
7.1.1	指针数据类型	179	8.1.4	结构体的初始化	229
7.1.2	指针运算符&和*的使用..	180	8.2	结构体数组.....	230
7.2	空间的动态分配与指针运算	181	8.3	结构体与指针	231
7.2.1	程序示例	181	8.3.1	指向结构体的指针	231
7.2.2	空指针	182	8.3.2	结构体中的成员包含 指针	234
7.2.3	存储器申请.....	182	8.3.3	用结构体类型指针 建立链表	235
7.2.4	存储器释放.....	184	8.4	结构体与函数	240
7.2.5	指针值的算术运算	184	8.4.1	结构体数据作为函数 的参数	240
7.3	指针与函数.....	187	8.4.2	返回指向结构体的指针 的函数	241
7.3.1	形参的数据类型是 指针类型	187	8.5	联合体	243
7.3.2	返回指针值的函数	190	8.5.1	问题提出与程序示例.....	243
7.3.3	指向函数的指针	191	8.5.2	联合体的声明和联合体 变量的定义.....	244
7.4	指针与一维数组	195	8.5.3	联合体变量成员的引用..	245
7.4.1	问题提出与程序示例	195	8.5.4	指向联合体变量的指针...	246
7.4.2	数组名及指针作为 函数参数	197			
7.4.3	指针与字符串	199			
7.5	二级指针	201			
7.6	指针数组	203			

8.5.5 联合体变量与函数	247	9.3.1 fputc()函数与 fgetc()函数..	269
8.6 枚举	249	9.3.2 fprintf()函数与 fscanf()函数	273
8.6.1 枚举的声明和枚举变量		273
的定义	249	9.3.3 fread()函数与 fwrite()函数	275
8.6.2 枚举变量的使用	249	9.3.4 fgets()和 fputs()	279
8.7 类型定义	250	9.4 文件的定位	280
8.8 程序举例	251	9.4.1 文件的顺序存取和	
8.9 典型错误及典型例题	254	随机存取	280
小结	259	9.4.2 rewind()函数	280
习题	260	9.4.3 fseek()函数	281
第 9 章 文件	263	9.5 程序举例	282
9.1 问题提出与程序示例	263	小结	287
9.2 文件操作的基本方法和		习题	287
相关概念	264	第 10 章 综合案例	290
9.2.1 数据文件	264	附录 A ASCII 码与字符对照表	300
9.2.2 文件类型指针	264	附录 B 运算符的优先级和结合性	302
9.2.3 文件的打开	265	附录 C printf()函数的转换模式及说明	304
9.2.4 文件的关闭	267	附录 D 预处理命令的使用	306
9.2.5 文件操作顺序	268	参考文献	310
9.2.6 C 语言的设备文件	269		
9.3 文件的读/写操作	269		

第 1 章 | C 语言概述

C 语言是一种通用的程序设计语言，它具有丰富的运算符和表达式，以及先进的控制结构和数据结构。C 语言具有表达能力强、编译目标文件质量高、语言简单灵活、容易移植、容易实现等优点。

1.1 程序与程序设计语言

1.1.1 程序

随着计算机走入寻常百姓家，“程序”已经不再是计算机科学使用的专用词汇了。在日常生活中，我们其实在不断地编写程序并执行，只不过并没有明确地意识到而已。例如，晚饭想吃传统食物饺子，应该怎么做呢？尽管简单，我们还是按照一般人的习惯来描述一下。

第一步，和面（要醒面，最好放在开始做）；

第二步，洗菜；

第三步，剁菜；

第四步，和馅（菜与现成的肉馅混合）；

第五步，擀皮；

第六步，包饺子；

第七步，下饺子；

第八步，吃饺子。

上面所描述的 8 个步骤，就是人们包饺子到吃饺子的“程序”。也许不同的人做的步骤并不完全一样，例如将第一步和第二步互换一下，也同样能把饺子包出来，所以干一件事的“程序”可以不唯一，这也是计算机程序的一个特点。

对于计算机来说，程序就是由计算机指令构成的序列。计算机按照程序中的指令逐条执行，就可以完成相应的操作。更准确一点，计算机执行由指令构成的程序，对提供的数据进行操作。计算机程序的操作对象是“数据”。这里的数据不是简单的阿拉伯数字，而是包括了各种现代计算机能够处理的字符、数字、声音、图像等。

实际上计算机自己不会做任何工作，它所做的工作都是由人们事先编好的程序来控制的。程序需要人来编写，使用的工具就是程序设计语言。

1.1.2 程序设计语言

目前,通用的计算机还不能识别自然语言,而只能识别特定的计算机语言。

计算机语言一般分为低级语言和高级语言。

低级语言直接依赖计算机硬件,不同的机型所使用的低级语言是完全不一样的。高级语言则不再依赖计算机硬件,用高级语言编写的程序可以方便地、几乎不加修改地用在不同类型的计算机上。

需要强调的是,无论采用何种语言来编写程序,程序在计算机上的执行都是由 CPU 所提供的机器指令来完成的。机器指令是用二进制表示的指令集。每种类型的 CPU 都有与之对应的指令集。

1. 低级语言

低级语言包括机器语言和汇编语言。

直接使用二进制表示的指令来编程的语言就是机器语言。使用机器语言编写程序时,必须准确无误地牢记每一条指令的二进制编码,才能编写程序。如果程序员面对的是“101110001110100000000011”这样的编码序列,能不头痛吗?而且,有时还要求把这些二进制编码再转换成八进制或十六进制数才能输入计算机,这不但加大了程序员的工作量,而且还增加了程序出错的概率。将大量的二进制编码序列准确地转换成八进制或十六进制数,不是一件容易的事。

机器语言的优点是执行速度快,并且可以直接对硬件进行操作,例如主板上的 BIOS 及一些设备的驱动程序等。

机器语言的缺点也是显而易见的。首先是可读性差,就是编写程序语句“101110001110100000000011”的人也未必马上就能看懂该句表示的是什么命令;其次,是可维护性差,别的程序员编写的程序(甚至是程序员自己编写的)很难看懂;再者,就是可移植性差,因为不同的机型有自己的一套机器指令,与其他机型的机器指令不兼容。另外,用机器语言编写程序的生产效率低下,并且不能保证程序有好的质量。

为了能够更方便地编写程序,人们用一些符号和简单的语法来表示机器指令,这就是汇编语言。例如,“101110001110100000000011”用汇编语言表示就是“MOV AX,1000”,该指令的功能是“将 1000 送入寄存器 AX 中”,是不是清楚多了?但是 CPU 并不能识别汇编语言,因此,需要一个“翻译”程序将汇编语言翻译成机器语言,我们把这种将汇编语言翻译成机器语言的程序叫作“汇编器”。汇编语言与机器语言的指令是一一对应的,所以,除了提高了一些可读性,汇编语言从根本上并没有改变机器语言的特点。可以说,汇编语言是面向机器语言的。当然,汇编语言也仍然具备机器语言的优点。许多大型系统(例如操作系统)的核心部分都是用汇编语言写的,因为这部分工作需要很高的效率,直接和硬件打交道。

那么,有没有办法真正提高程序的可读性,可维护性和可移植性呢?回答是肯定的,就是使用高级语言。

2. 高级语言

高级语言是一种比较接近自然语言和数学语言的程序设计语言。高级语言的出现大大提高了程序员的工作效率,降低了程序设计的难度,并改善了程序的质量。用高级语言编写的程序看起来更像是英语,很容易读懂,不但使程序具备良好的可读性和可维护性,而且使更多的人掌握了程序设计方法,从而使计算机技术得到迅速的应用和普及。